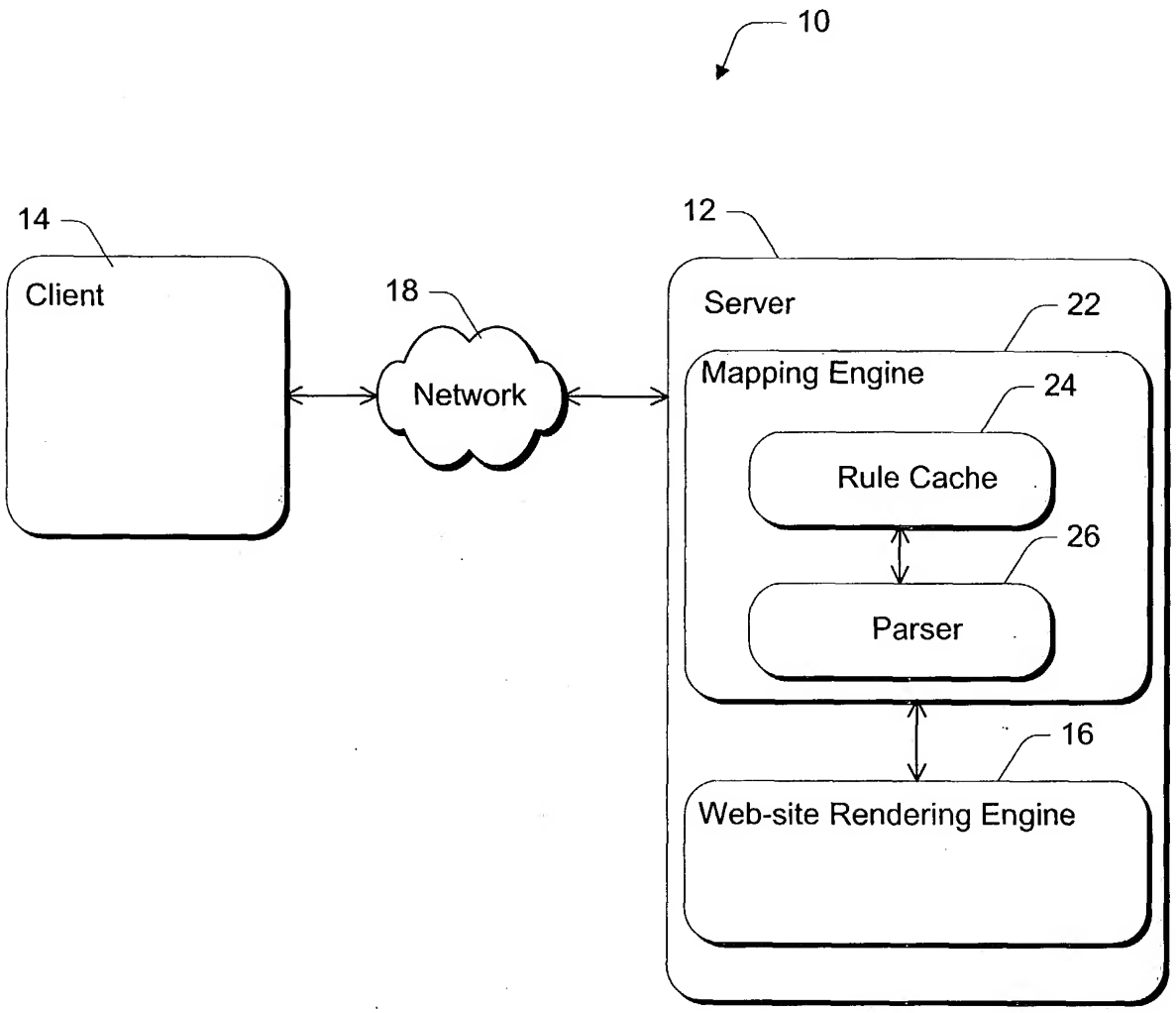
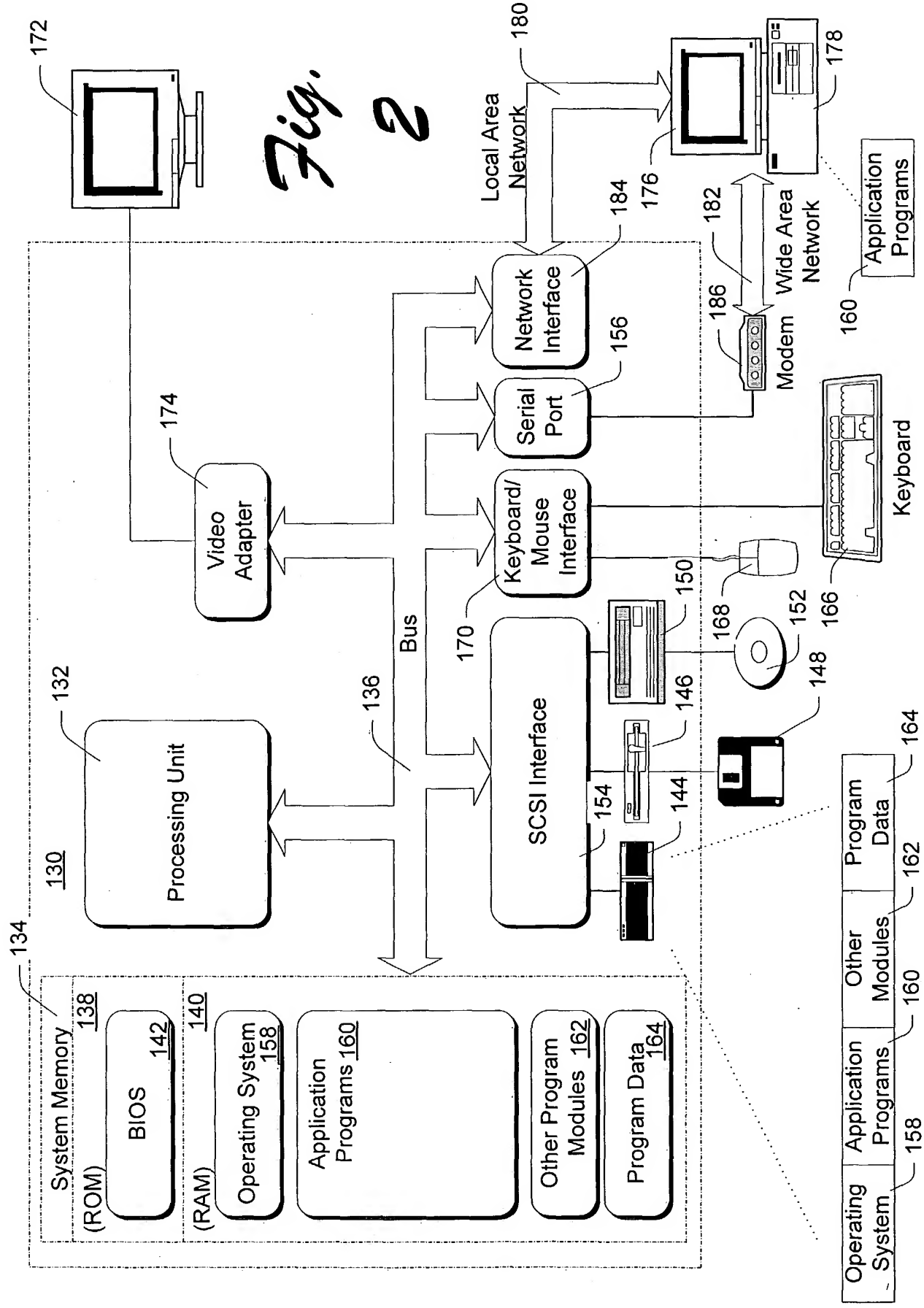

10
14
Client
18
Network
12
Server
22
Mapping Engine
24
Rule Cache
26
Parser
16
Web-site Rendering Engine

Fig. 1

Fig. 2


172
180
Local Area Network
178
176
184
182
Wide Area Network
Application Programs
160
Network Interface
186
Modem
156
Serial Port
174
Video Adapter
Keyboard/ Mouse Interface
Keyboard
Bus
150
168
152
166
170
SCSI Interface
146
148
132
136
Processing Unit
154
144
130
134
System Memory
138
(ROM)
BIOS
142
140
(RAM)
Operating System
158
Application Programs
160
Other Program Modules
162
Program Data
164
Other Modules
Program Data
Application Programs
Operating System

Regular Expression Escape Characters:

`+` - At least one or more of the preceding characters or expressions. For example, ba+c matches bac, baac, but not bc.

`[]` - Any one of the characters contained in the brackets, or any of an ASCII range of characters separated by a hyphen `(-)`. For example, b[aeiou]d matches bad, bed, bid, bod, and bud (but not bead or beed) and r[eo]+d matches red, rod, reed, rood, reod, roed, reood, roeod, etc.

`^` - The beginning of a line

`$` - The end of a line

`*` - None or more of the preceding characters or expressions. For example, ba*c matches bc, bac, baac and so on.

`[^]` - Any character except those following the caret (^) character in the brackets, or any of an ASCII range of characters separated by a hyphen `(-)`. For example, `x[^0-9]` matches `xa, xb, xc,` and so on, but not `x0, x1, x2,` and so on.

`()` - Indicates a tagged expression to retain for replacement purposes. Each occurrence of a tagged expression is numbered according to its order and its replacement expression is \n, where 1 corresponds to the first tagged expression, 2 to the second, and so on.

`()\:n` - *(n is an integer between 1and 255)* Indicates a tagged expression (as above) that needs to be used as the n^{th}-input parameter for an external LookUp() function.

`{c!c}` - Any one of the characters separated by the alternation symbol `(\!)`. For example, `{j!u}+fruit` matches `jfruit, ufruit, jjfruit, ujfruit` etc.

`{}` - Any sequence of characters between the escaped braces. For example, `{ju}+fruit` matches `jufruit, jujufruit` but not `ufruit, jfruit or ujfruit`

`\:a` - Any single alphanumeric character `[a-zA-Z0-9]`

`\:c` - Any single alphabetic character

`\:d` - Any decimal digit `[0-9]`

`\:z` - Any unsigned decimal integer `[0-9]+`

`\:h` - Any hexadecimal number `[0-9a-fA-F]+`

`\:w` - Any alphabetic string `[a-zA-Z]+`. The string need not be bounded by white spaces or appear at the beginning or the end of a line

`\:q` - Any quoted string `{"[^"]*!'[^']*'}`

`\N` - (*N is a digit*) The sub-string in the target that matched the nth-tagged expression

`\` - Escape character to remove the pattern match characteristics from the special characters listed above. For example, 100$ matches 100 at the end of a line, but 100\$ matches the character string 100$ anywhere on a line.

Fig. 3

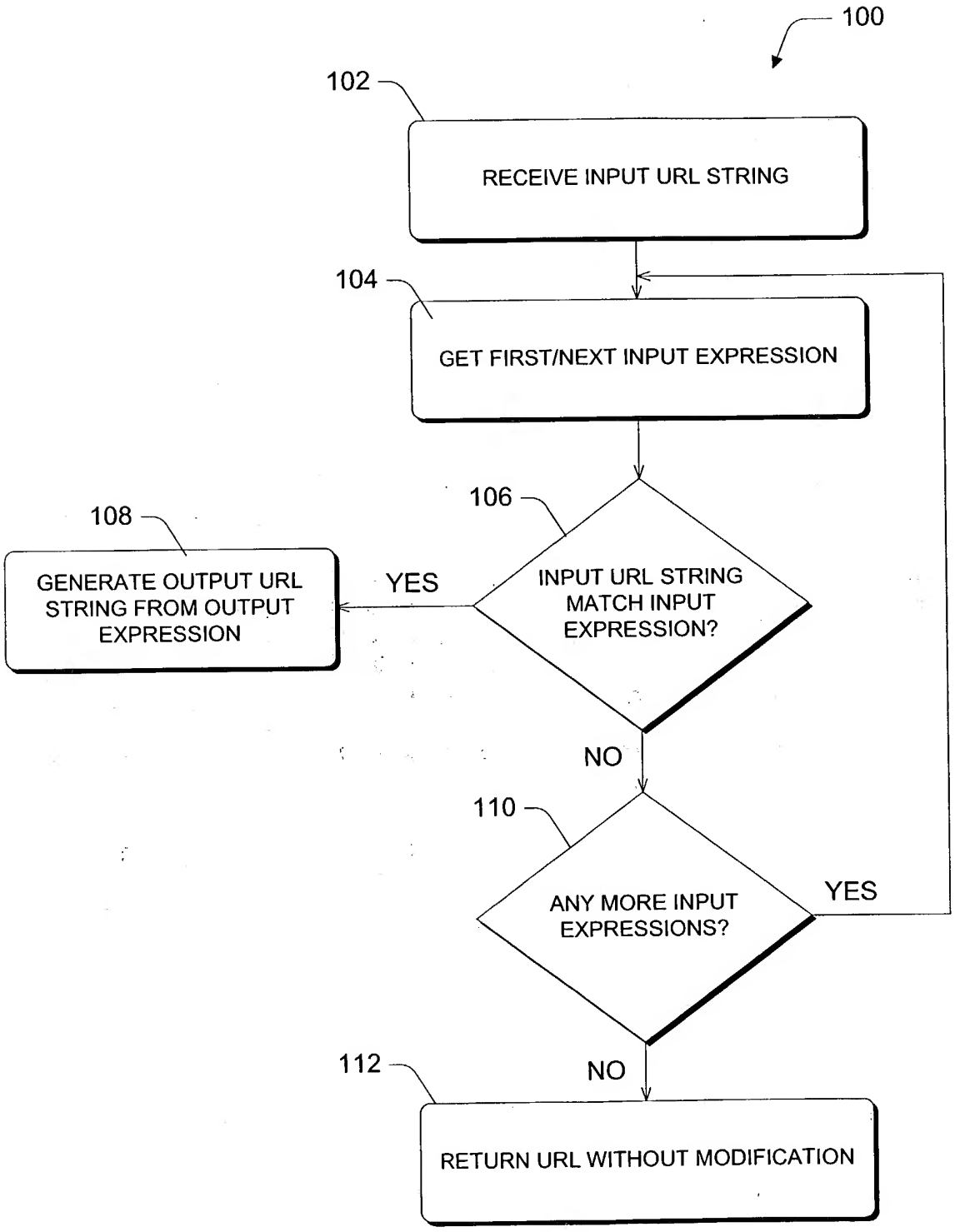

100
102
RECEIVE INPUT URL STRING
104
GET FIRST/NEXT INPUT EXPRESSION
106
INPUT URL STRING MATCH INPUT EXPRESSION?
YES
108
GENERATE OUTPUT URL STRING FROM OUTPUT EXPRESSION
NO
110
ANY MORE INPUT EXPRESSIONS?
YES
NO
112
RETURN URL WITHOUT MODIFICATION

Fig. 4

Group ID = 1, Group Tag = Match One, Group bit-mask = 0x01

Rule ID	Rule Action Type	Input Expression	Output Expression
1	N	^Sidewalk.com/([^/]+)	/scripts/city.dll?city=\1
2	N	^Sidewalk.com:/([^/]+)	/scripts/city.dll?city=\1
3	N	^Sidewalk.com:80/([^/]+)	/scripts/city.dll?city=\1
4	N	^([^.]+)/([^/]+)	/scripts/city.dll?city=\2
5	N	^([^.]+):/([^/]+)	/scripts/city.dll?city=\2
6	N	^([^.]+):80/([^/]+)	/scripts/city.dll?city=\2
7	N	^([^.]+).sidewalk.com	/scripts/city.dll?city=\1
8	N	^([^.]+).sidewalk.com:	/scripts/city.dll?city=\1
9	N	^([^.]+).sidewalk.com:80	/scripts/city.dll?city=\1
10	N	^Www.([^.]+).sidewalk.com	/scripts/city.dll?city=\1
11	N	^Www.([^.]+).sidewalk.com:	/scripts/city.dll?city=\1
12	N	^Www.([^.]+).sidewalk.com:80	/scripts/city.dll?city=\1

Fig. 5

Group ID = 2, Group Tag = Match One, Group bit-mask = 0x02

Rule ID	Rule Action Type	Input Expression	Output Expression
1	N	/$	&LID=1
2	N	/link/(\z)$	&LID=\1
3	N	/link/(\z)/$	&LID=\1
4	N	/link/(\z)/(\w)$	&\2&LID=\1
5	N	/link/(\z)?(\w)$	&\2&LID=\1
6	N	/detail/(\z)$	&EID=\1
7	N	/detail/(\z)/$	&EID=\1
8	N	/detail/(\z)/(\w)$	&\2&EID=\1
9	N	/detail/(\z)?(\w)$	&\2&EID=\1
10	N	/([^/]+)\:1$	&EID=\(frlkup, 1)
11	N	/([^/]+)\:1/$	&EID=\(frlkup, 1)
12	N	/([^/]+)\:1/(\w)$	&\2&EID=\(frlkup, 1)
13	N	/([^/]+)\:1?(\w)$	&\2&EID=\(frlkup, 1)

Fig. 6

Rule ID	Rule Action type	Input Expression	Output Expression
1	N	^Sidewalk.com/([^/]+)\:1	city.dll?scopeid=\(scope, 1)
2	N	^Sidewalk.com:/([^/]+)\:1	city.dll?scopeid=\(scope, 1)
3	N	^Sidewalk.com:80/([^/]+)\:1	city.dll?scopeid=\(scope, 1)
4	N	^([^.]+)/([^/]+)\:1	city.dll?scopeid=\(scope, 1)
5	N	^([^.]+):/([^/]+)\:1	city.dll?scopeid=\(scope, 1)
6	N	^([^.]+):80/([^/]+)\:1	city.dll?scopeid=\(scope, 1)
7	N	^([^.]+)\:1.sidewalk.com	city.dll?scopeid=\(scope, 1)
8	N	^([^.]+)\:1.sidewalk.com:	City.dll?scopeid=\(scope, 1)
9	N	^([^.]+)\:1.sidewalk.com:80	City.dll?scopeid=\(scope, 1)
10	N	^Www.([^.]+)\:1.sidewalk.com	City.dll?scopeid=\(scope, 1)
11	N	^Www.([^.]+)\:1.sidewalk.com:	City.dll?scopeid=\(scope, 1)
12	N	^Www.([^.]+)\:1.sidewalk.com:80	City.dll?scopeid=\(scope, 1)

Fig. 7